Xindong Duan[1*]

Research and Application of Network Traffic Anomaly Detection Algorithm Based on Deep Learning

Journal of Electrical Systems

Abstract: - Traffic anomaly detection is the process of spotting odd trends or departures from typical network traffic behavior. One drawback of traffic anomaly detection systems is their susceptibility to false positives. These systems may sometimes incorrectly flag normal variations in traffic patterns as anomalies, leading to unnecessary alerts and potentially diverting resources towards investigating non-existent issues. In this manuscript Research and Application of Network Traffic Anomaly Detection Algorithm Based on Deep Learning (RA-NTADA-EPTANN) is proposed. Initially, the data are collected from DS2OS Dataset. The collected data are fed to Pre-processing segment. In pre-processing segment, Confidence Partitioning Sampling Filtering (CPSF) is used to data cleaning, handling the missing values and noisy data. Then, the pre-processed data are given to feature selection process. Feature selection is done by Humboldt squid optimization algorithm (HSOA). In feature selection technique, seven features are selected. Finally the selected feature attributes are given to efficient predefined time adaptive neural network (EPTANN)for Anomaly Detection classifying such as the DOS attack, data probing, malicious control, malicious operation, scan, spying, and wrong setup. In general, Efficient Predefined Time Adaptive Neural Network(ANN) does not express some adaption of optimization strategies for determining optimal parameters to ensure accurate Anomaly Detection. Hence, Multi-Agent Cubature Kalman Optimizer (MACKO)is to optimize to Efficient Predefined Time Adaptive Neural Network which accurately Anomaly Detection. The proposed technique implemented in python and efficacy of RA- NTADA-EPTANN technique is assessed with support of numerous performances like Accuracy, Computational Time, F1-Score, Precision, Recall and ROC is analysed. Proposed RA- NTADA-EPTANN method attains 15.12%, 22.23% and 35.32% higher computational time analysed with the existing for Deep neural network based anomaly detection in Internet of Things network traffic tracking for the applications of future smart cities (AD-IOTNTT-DNN),Network traffic anomaly detection method based on chaotic neural network (NTAD-CNN) and Deep learning-based network anomaly detection and classification in an imbalanced cloud environment (NAD-ICE-DCNN),respectively.

Keywords: Confidence Partitioning Sampling Filtering, Efficient Predefined Time Adaptive Neural Network, Humboldt squid optimization algorithm, Multi-Agent Cubature Kalman Optimizer, Network Traffic Anomaly Detection.

I. INTRODUCTION

Traffic anomaly recognition is a crucial constituent of network security and management systems, tasked with identifying abnormal patterns or behavior within network traffic [1].Anomaly detection systems have the ability to differentiate between normal traffic patterns and potentially harmful activities such as denial-of-service attacks, malware propagation, or unauthorized access attempts. They achieve this by employing a range of techniques, including statistical analysis, machine learning(ML) algorithms, and signature-based detection [2]. These systems continuously monitor network traffic in real-time, flagging any abnormalities that can point to a security breach or operational problem by comparing the present behavior with previous data or predefined criteria [3]. In addition to improving network security, efficient anomaly detection helps to maximize network efficiency by quickly detecting and addressing any anomalies that could compromise the network's overall dependability and functionality [4].Through the network, cloud computing offers simple access to a range of programmable and flexible computer resources in response to customer demand [5]. Cloud computing services are obtainable through standard network and internet protocols (IP). Apart from the distinct advantages of cloud computing, it is imperative to acknowledge the existence of unsecured communication and cloud network threats [6]. There are numerous methods for addressing network intrusions.Due to its ever-expanding scope, the Internet environment is vulnerable to a wide range of more frequent intrusion attacks, endangering the data and assets of both individuals and businesses [7].Reliable network connections are necessary for the widespread use of cloud computing resources, and irregularities in network traffic have a detrimental effect on the availability of cloud resources [8]. These network anomalies must be found and predicted using anomaly detection technologies. The use of ML techniques in anomaly detection tool implementations has grown in prominence in recent years [9]. Developers of anomaly detection software and managers of cloud infrastructures have a wide range of potential solutions at their disposal. They can choose from various machine learning methods to

1 [1*]School of Digital Media and Design Arts, Nanyang Institute of Technology, Nanyang, Henan, 473000, China

[1*]Email: xindongduanphd@gmail.com

address the different types of network anomalies. [10].The landscape of network attack technology is constantly evolving and advancing, driven by the rapid emergence of new technologies such as big data, cloud computing, and the growing Internet of Things (IOT) [11].

As a result, network attack detection technology must also evolve iteratively. These technologies are linked to three primary issues: the inconsistent network attack samples, The representation of complex and heterogeneous network traffic data, and the ongoing evolution of assaults, pose a challenge for anomaly detection models in terms of accuracy [12]. Researchers have developed several methods for detecting network attacks using deep learning (DL) techniques to address these issues. Continuous monitoring of real-world traffic surveillance footage is essential to promptly detect incidents and take appropriate action in the event of a fatality [13]. But having a human oversee them all the time is laborious and prone to mistakes. Consequently, by framing the issue as anomaly detection, a deep learning method for the automatic detection and localization of traffic accidents has been developed [14].Whether it's a smart meter, smart home, smart industry, or smart irrigation system, everything is becoming smarter in this day and age. The term smart here refers to the use of the IoT [15]. Ansignificant domain of an IoT is espionage, and the growing usage of IoT infrastructure in these fields has resulted in an increase in threats, attacks, irregularities, and node failure. Recently, energy efficiency and carbon emission reduction are major concerns in many businesses, and sustainability is at the heart of green technologies [16]. The frequency of cyberattacks generating sustainability problems in industries is rising along with this worry. Industrial systems that regulate and keep an eye on the proper operation of systems and processes are impacted by these cyberattacks [17]. Moreover, they are exceedingly specialized, hidden from conventional cyber security solutions, and necessitate familiarity with the target industrial processes. To sum up, traffic anomaly detection is essential to maintaining the effectiveness, security, and stability of contemporary networks [18]. The utilization of sophisticated algorithms, machine learning methodologies, and real-time monitoring systems facilitates prompt detection and resolution of anomalous patterns, possible hazards, and malfunctions in performance [19]. Robust anomaly detection systems are crucial as network infrastructures becoming more sophisticated and technology advances. Their significance cannot be emphasized. We may further improve the capabilities of traffic anomaly detection systems by ongoing innovation and collaboration between researchers, engineers, and industry stakeholders, ultimately leading to safer, more dependable, and resilient networks in the future [20].

The high computational complexity and resource needs of deep learning-based network traffic anomaly detection algorithms are frequently criticized, despite the fact that they show promise in detecting intricate patterns and irregularities in network data. Deep learning models can be difficult to apply in situations with limited resources or in real-time systems since they often require vast volumes of data for training and significant computer capacity for inference. Furthermore, the black-box aspect of deep neural networks may impede these algorithms' interpretability and make it challenging to comprehend the reasoning behind particular conclusions, which is essential for efficient cyber security analysis and decision-making. Therefore, the actual implementation of deep learning-based algorithms may encounter challenges because of their processing needs and interpretability concerns, even though they are effective in detecting network anomalies.

Using an effective predetermined time adaptive neural network architecture is one of the unique ways to network traffic anomaly detection. Unlike traditional methods that rely on fixed architectures, this approach dynamically adjusts its network structure based on the temporal characteristics of network traffic data. By incorporating adaptability into the neural network's design, the model can effectively capture subtle changes and evolving patterns in network behavior, thus enhancing its ability to detect anomalies with high accuracy and efficiency. This adaptability enables the algorithm to autonomously optimize its architecture in response to varying traffic conditions, leading to improved performance in real-world scenarios where network dynamics are constantly changing. Additionally, the use of DLtechniques facilitates the extraction of intricate features from raw traffic data, further enhancing the algorithm's capability to discern between normal and anomalous network activities. Overall, the integration of efficient predefined time adaptive neural networks presents a promising avenue for advancing the field of network traffic anomaly detection, offering heightened adaptability and robustness in detecting emerging threats and abnormalities in complex network environments.

Major contribution of this research work summarized as below,

- In this manuscript RA-NTADA-EPTANN)

- The dataset sourced from DS2OS was employed in conducting the pre-processing using Confidence Partitioning Sampling Filtering (CPSF), following Using the Humboldt Squid Optimization Algorithm, pre-processed data are supplied into the feature selection process (HSOA).
- Anomaly detection, data probing, malicious control, malicious operation, scan, espionage, and incorrect setup are all classified using the Efficient Predefined Time Adaptive Neural Network (EPTANN).Finally accurately classifying the defect by using Multi-Agent Cubature Kalman Optimizer (MACKO).

Remaining portions of this work are arranged as below: part 2 analyses literature review, part 3 describes proposed method; part 4 illustrates outcomes; part 5 presents conclusion.

II. LITERATURE SURVEY

Numerous research related to Network Traffic Anomaly Detection have previously been published in the literature; a selection of those studies is examined below.

Reddy et al. [21] have suggested a Deep neural network-based anomaly detection in Internet of Things (IoT) network traffic tracking will be essential for future applications of smart cities. This research focused on the full trial ability investigation and evaluations on deep learning neural network architecture for the detection of seven attack types contained in the traffic traces data set of the Distributed Smart Space Orchestration System (DS2OS). The simulation algorithm's empirical findings show that deep neural network design functions effectively, showing noticeable gains in the majority of categorical attacks. It attains higher accuracy and lower precision.

Sheng and Wang, [22] have suggesteda chaotic neural network-based technique for detecting anomalies in network traffic. The identification of network aberrant traffic was a common topic in this article on network security. In order to solve the problems of high dimensional anomalous traffic and overfitting of the classification model owing to outliers, this research develops a network traffic anomaly detection model based on the concept of chaotic neural networks. The model first uses the degree of correlation between features and class tags to determine mutual information. Next, in order to extract relevant characteristics for the purpose of identifying abnormalities in internet traffic, it selects a superior feature subset. A chaotic neural network approach was then used to find the ideal feature subset from the original feature set. An adaptive strategy was then used to add and delete the iterated features. It provides higher computation time and lower accuracy.

Vibhute and Nakum, [23] have suggesteda detection and categorization of network anomalies using deep learning in an unbalanced cloud environment. In this research, we show that end-to-end system communication has significantly increased as a result of advancements in computer networking. But security concerns have also been brought up. As such, it remains difficult to identify anomalies in a complicated cloud environment. Consequently, the deep Convolutional Neural Network (CNN) model was presented in this article as a means of identifying and categorizing network intrusions from an unbalanced cloud environment in near real time. In order to choose the most appropriate characteristics to feed into the CNN model, the random forest model was also provided and used. It provides higher precision and low computation time.

Ajila et al. [24] have suggestedan examination of ML techniques based on errors for the identification and classification of network anomalies. In this research, we conduct experiments to identify the optimal error-based ML method in terms of accuracy and build time for both anomaly detection and anomaly attack categorization. A two-stage framework for anomaly and categorization has been developed for experimental assessment. The type of attack is determined by the second stage if the results of the first stage are atypical. Whether a network flow was normal or aberrant is determined at the first step. The ultimate goal was to use the best algorithm for an online stream model of network intrusion detection.In order to achieve this, four sets of experiments are established, five research proposals are defined, and four research questions are posed.It attains higher F1- score and lower precision.

Hooshmand and Hosahalli, [25] have suggestedDL algorithms are utilized for the identification of network anomalies. The study provides a model that adapts this performance to the task of network anomaly detection in cyber-security using a one-dimensional CNN architecture. In the first step, the authors' technique divides the data from network traffic into three groups: OTHER protocol categories, user datagram protocol(UDP), and transmission control protocol(TCP). After then, each category was managed separately. In order to solve the issue of class imbalance, the synthetic minority over-sampling approach was employed for over-sampling and

the Chi-square technique for feature selection prior to training the model. Convolutional neural networks(CNN), or CNNs, are artificial neural networks(ANN) with a unique design that are feed-forward. It was the de facto standard for a number of ML and computer vision procedures. It achieves reduced accuracy and greater recall.

Patil et al. [26] have suggesteda PCA and BiGAN-based network traffic anomaly detection system. This research offered a lightweight and intelligent system for detecting unusual network traffic. The system largely used principal component analysis for feature extraction and dimensionality reduction, and it uses a bidirectional generative adversarial network (BiGAN) model to identify aberrant network traffic. The proposed framework was evaluated and compared with existing DL models using the KDDCUP-99 dataset. To comprehend the peculiarities of the dataset and to depict the KDDCUP-99 dataset, a variety of visualization techniques were also used. Our research demonstrates how crucial feature reduction is to raising the BiGAN models' overall effectiveness. It achieves a lower F1-score and more accuracy.

Yao et al. [27] have suggested a Principal Component Analysis and DNN-Based Traffic Anomaly Detection in Wireless Sensor Networks (WSN). Based on principal component analysis (PCA) and a DCNN, this paper proposes a method for identifying DoS traffic abnormalities in WSNs, which are vulnerable to assaults and have limited storage space on their devices. The proposed model, which is lighter than the conventional deep learning structure and has more powerful feature extraction capabilities, may be able to detect aberrant network traffic in WSNs devices with limited storage capacity. Researching network intrusion prevention for wireless sensor networks was essential because of the growing popularity of wireless networks, which has led to the rapid advancement of WSNs and increased security concerns due to their flexibility and ease of implementation. A typical kind of network assault was denial of service (DoS), which takes down the target network in order to achieve its objective. DoS attacks would be lethal against WSNs devices with constrained resources. It provides higher accuracy and lower precision.

III. PROPOSED METHODOLOGY

The proposed RA-NTADA-EPTANN is discussed in this section. Block diagram of proposed RA-NTADA-EPTANN is illustrated in figure 1.It includes DS2OS dataset, pre-processing using Trainable Kalman Filter used to Data cleaning, handling missing values and noisy data, after then the pre-processed data are given to Feature Selection Using Humboldt squid optimization algorithm, These features are then organized into a feature vector. Using an efficient predefined time adaptive neural network, the final anomaly detection method classified DOS attacks, data probing, malicious control, malicious operation, scans, surveillance, and incorrect setups. optimizing using Multi-Agent Cubature Kalman Optimizer are processes that make up this procedure. Consequently, a full explanation of each stage is provided below,

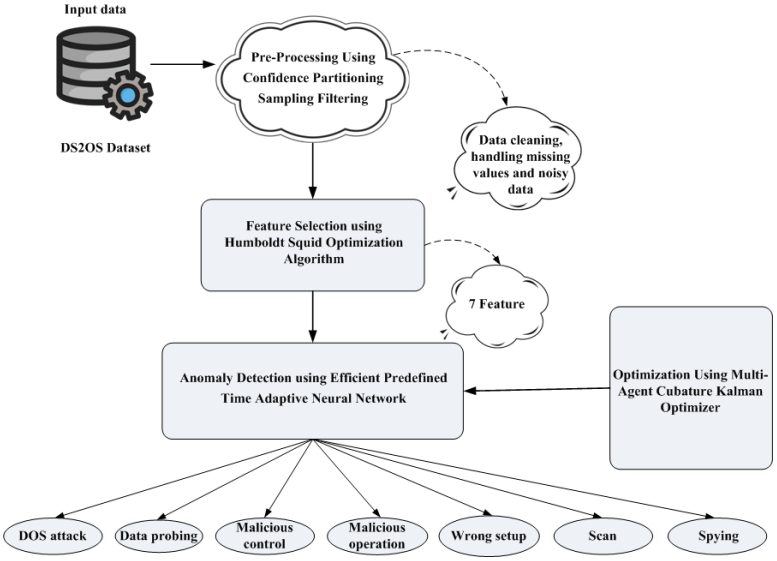

Figure 1: Block diagram of RA-NTADA-EPTANN

A. Data Acquisition

The data is gathered from the DS2OS dataset [21]. Because they originate from the application layer, they are very different from conventional network traces. An excellent resource for assessing the effectiveness of AI-

based cyber security plans for smart cities, smart businesses, and several other IoT applications is this open-source dataset. A total of 357952 samples make up the DS2OS; of them, 347935 are categorized as normal samples and 10017 as aberrant results. There are 8 classes and 13 characteristics in this collection. A thorough breakdown of the DS2OS class distribution is provided. The DS2OS Features are shown in Table 1.

Table 1: DS2OS Features

SI.No	Feature Name	Description	SI.No	Feature Name	Description
1	Source	Nominal	8	Accessed Node Address	Nominal
2	Source Address	Nominal	9	Accessed Node Type	Nominal
3	Source Type	Nominal	10	Operation	Nominal
4	Source Location	Nominal	11	Value	Continuous
5	Destination Service Address	Nominal	12	Timestamp	Discrete
6	Destination Service Type	Nominal	13	Normality	Nominal
7	Destination Location	Nominal			

One of the main security concerns with the Industrial Internet of Things (IIoT) is anomaly detection since cyberattacks are becoming more frequent and dangerous for dispersed devices and critical infrastructure networks.

B. Pre-Processing Using Confidence Partitioning Sampling Filtering

In this section, the input data are pre-processed utilizing CPSF [28]. In pre-processing segment is used to data cleaning, handling the missing values and noisy data. CPSF is a methodology used in various fields, notably data analysis and machine learning, to enhance the efficiency and accuracy of processes involving large datasets. A number of benefits flow from CPSF, which systematically divides data into confidence levels for targeted sampling and filtering. First off, by concentrating efforts on data subsets with more confidence, it maximizes computational resources by cutting down on processing expenses and time. Second, by giving high-confidence data point's priority, it improves forecast accuracy and raises the caliber of models and insights produced. Furthermore, CPSF makes it easier to recognize and control the uncertainty included in complicated datasets, which improves risk assessment and decision-making. All things considered, CPSF gives practitioners the ability to effectively extract relevant information from massive amounts of data, empowering them to make defensible judgments and draw actionable insights. The method of choice, Confidence Partitioning Sampling Filtering (CPSF), was selected because of its ability to handle big datasets with resilience and maintain representative sampling. CPSF enables targeted sampling by dividing the data according to confidence levels. This concentrates resources on places where the data has low confidence, enhancing the overall quality of analysis without compromising efficiency. This approach is the best option in situations when accuracy and scalability are critical since it not only maximizes computational resources but also improves the reliability of outcomes.Let $p(x)$ be the PDF of a distribution, and let $C^0_{p(x)} \cdot C^{\alpha}_{p(x)}$ be its bounded subspace, which satisfies

$$s.t \int_{C^{\alpha}_{P(x)}} P(x)dx = 1-\alpha \tag{1}$$

Where $s = \int_{C^{\alpha}_{P(x)}} 1dx, 0 \le \alpha \le 1.$ Then $C^{\alpha}_{p(x)}$ is called the $p(x)$ confidence $1-\alpha, s$ is the represent a parameter associated with the segmentation of the signal, t is the typically represents time in signal processing, $p(x)$ is could denote a function that depends on the variable x. dx is typically represents an Infinitesimal change in the variable x .CPSF is a approach employed in network traffic analysis to detect anomalies.

$$\omega \equiv [\omega_1, \omega_2, \ldots \omega_L]^T \tag{2}$$

$$\omega l = \frac{p(\hat{x}l)}{\sum_{l=1}^{L} p(\hat{x}l)} \tag{3}$$

Where l is represent a length or distance parameter. ω is the angular frequency in radians per second. L is representing a length or size parameter. T is represents a period of time interval in signal processing. It involves partitioning network traffic data into smaller segments based on confidence levels, then sampling and filtering these segments for abnormal patterns.

$$p(x)\, \alpha \lim_{\alpha, \tau \to 0} \sum_{l=1}^{+\infty} \omega l \delta(x - \hat{x}l) \tag{4}$$

In this case, L blocks are formed using τ τ, and the dimension is $\tau=[\tau_1,\tau_2,..,\tau_D]^T$. D. Samples $\hat{X}$, $\hat{X}\equiv[\hat{x}_1,\hat{x}_2,...\hat{x}_L]^T$ .are obtained by deploying ten samples in the middle of each block. This strategy concentrates on segments that have a higher likelihood of harboring abnormalities in an effort to maximize anomaly identification. Each sample's weight is determined using the formula below:

$$p(x)\,\alpha\,\sum_{l=1}^{L}\omega l\,\delta(x-\hat{x}l) \tag{5}$$

CPSF can detect known and unknown anomalies and adapt to changing network conditions by dynamically modifying confidence levels. By offering early identification of suspicious activity, such as DDoS attacks or atypical traffic patterns suggestive of cyber dangers, the program improves network security. Finally, CPSF filter is removed data cleaning, handling the missing values and noisy data, these pre-processed data are fed into feature selection.

C. Feature Selection using Humboldt Squid Optimization Algorithm

The features selection depends on HSOA [29]. HSOA is Humboldt squid collective behavior serves as an inspiration for this novel and effective technique to problem solving. Its capacity to successfully explore intricate and dynamic search spaces particularly in optimization problems is its main asset. Faster convergence and higher-quality solutions are achieved by HSOA, which balances exploration and exploitation in an optimal manner by imitating the swarming activity of these extremely adaptive animals. In addition, it is a promising tool for addressing a extensive range of optimization difficulties in a variety of industries, from engineering and economics to biology and beyond, because to its scalability, simplicity of implementation, and flexibility to multiple problem domains. The Humboldt squid is a highly intelligent and adaptive marine creature, and its collective behavior and hunting techniques serve as inspiration for the Humboldt Squid Optimization Algorithm (HSOA). This algorithm is unique in that it mimics the cooperative and swarming characteristics of these cephalopods, making it an effective means of handling optimization problems. Similar to how Humboldt squid cooperate to find and seize prey in the wide ocean, HSOA makes use of the idea of teamwork among individuals to traverse challenging search spaces and get to the best answers. HSOA provides a promising method for addressing a variety of optimization issues in a diversity of fields, including engineering, finance, medicine, and more, by mimicking the principles found in nature. Its ability to harness the power of collective intellect is what makes it so successful and captivating choice for researchers and practitioners seeking innovative optimization techniques.

1) Stepwise procedure of HSOA

A methodical process is outlined for choosing the best feature selection with the HSOA Algorithm. Given that it handles both the exploration and exploitation phases, HSOA is theoretically a global optimization method. The following is a mathematical breakdown of the steps in the proposed HSOA.

Step 1: Initialization

There are fish swarms and Humboldt squid in the HSOA population. The following pseudo code is used by HSOA to create the first population. As demonstrated, the top people fish make up the remaining population, with Humboldt squid being classified as such. The fact that blot squid are more physically fit and have larger bodies than school fish means that this problem is compatible with nature. Thus, it is given in equation (6)

$$y_i^j = y_{i,\min}^j + rand.\left(y_{i,\min}^j - y_{i,\min}^j\right) \tag{6}$$

Here, y_i^j stands for the j^{th} decision variable, which is used to determine the II i^{th} candidate's starting position; $rand$ is a random number with a uniform distribution inside the interval [0, 1]; $y_{i,\min}^j$ stands for the j^{th} candidate's lower bound; The higher boundaries of the j^{th} variable are represented by $y_{i,\min}^j$;

Step 2: Random Generation

Parameters for the input are created at random after startup. Their explicit hyperparameter condition determines the best fitness value to choose.

Step 3: Fitness Function

First, a solution candidate matrix representing the starlings starting positional vectors is identified. This matrix is first assigned to random values inside a search space; Equation is the fitness function that chooses the best characteristics(7)

$$Fitness\ function = \left[\ Selecting\ Optimal\ \ Features\ \right] \tag{7}$$

Step 4: Successful Attack for Optimizing

For the search operation, five mechanisms are specified in order to directly model this process. These processes include fish school attacks, fish escapes, successful attacks, bigger squid attacking smaller squids, and Humboldt squid mating. In HSOA, mating is transformed from exploration through repeated attacks by fish schools, stronger squids attacking smaller squids, and other means of searching. But fish escape manages exploration in each repetition. The existing position of the Humboldt squid is replaced with the new position of the squid after the new position for the fish and squid is changed. Thus, it is given in equation (8)

$$YS_j^d = \begin{cases} YS_i = YS_{new,i} & if\ G_{r,new\,j} < G_{S,j} \\ Successful\,escape, & otherwise \end{cases} \tag{8}$$

Here, YS_j^d is represent new location for Humboldt squid; $G_{r,new\,j}$ is represent the new fitness functions; $G_{S,j}$ is represent the current fitness function of the i^{th} Humboldt squid; after the squid's attack on the fish school, the fish escape to an area chosen at random.

Step 5: Attack of Stronger Squids to Smallest Squids For Optimizing

If the fish and Humboldt squid in the previous phases are unable to find a better place, it is assumed that there are no more fish to seek. Thus, the larger Humboldt squid consumes the smaller ones. At this point, the Humboldt squid's position, thus, it is given in equation (9)

$$YS_{new,j}^p = YS_{new,j}^p + P_{jet2.}\left(YS_{new,j}^p - Y_a^p\right) \tag{9}$$

Here, $P_{jet2.}$ is denotes the current position, Y_a^p represent the normal random vector,In HSOA, $YS_{new,j}^p$ stands for the second velocity parameter that is used to produce the egg location. It was originally used to improve the method for deferential evolution.

Step 6: Termination Condition

In order to pick the optimal features, HSOA is used. Step 3 will then be repeated repeatedly until the stopping point is reached. From the DS2OS dataset, HSOA chose seven characteristics. Next, the Efficient Predefined Time Adaptive Neural Network receives the characteristics that have been chosen. Table 2 demonstrates how HSOA chose the characteristics.

Table 2: HSOA chose the characteristics.

SI.NO	Features	SI.NO	Features
1	Source ID	5	Source Location
2	Destination Service Address	6	Destination Service Type
3	Destination Location	7	Operation
4	Timestamp		

D. Anomaly Detection using Efficient Predefined Time Adaptive Neural Network

In this section the Anomaly Detection using Efficient Predefined Time Adaptive Neural Network (EPTANN) [30] from feature selected data.Using the EPTANN, anomalies such denial-of-service attacks, data probing, malicious control, malicious operation, scanning, espionage, and incorrect configuration can be found. A paradigm shift in neural network architecture is provided by the Efficient Predefined Time Adaptive Neural Network (EPTANN), especially for time-series prediction problems. Predefined time intervals are a key component of EPTANN's architecture, which maximizes computational resources and speeds up training while maintaining predicted accuracy. This novel method simplifies existing recurrent networks and allows for smooth adaptation to different time intervals, which makes it a great option for real-time applications including anomaly detection, dynamic system control, and financial forecasting. As a leader in developing neural network skills in time-sensitive fields, EPTANN offers significant benefits in speed, resource efficiency, and adaptability because to its effective handling of temporal data.The selection of an EPTANN is based on its computational efficiency in handling the time-series data's dynamic characteristics. Conventional neural networks frequently exhibit subpar performance or computational bottlenecks as a result of their inability to effectively adapt to shifting temporal patterns. By adding predetermined time intervals, EPTANNs provide a solution by enabling targeted

network architecture modifications based on the temporal properties of the input. This method is appealing for jobs where flexibility to changing temporal dynamics are critical since it improves predicted accuracy while streamlining computational resources.

$$\dot{S}_{et}(h) = -\lambda_{et}(h)\Phi_{et}(F(h)) \quad (10)$$

where (h) is the definition of the variable at time, and $\dot{S}_{et}$ is the rate of change of a quantity over time. A parameter that regulates the pace of change is defined by h and λ . Φ_{et} defines the state variable, $F(h)$ defines the activation function, and Φ_{et} defines a parameter that regulates the pace of change. Effective Predetermined Time One kind of neural network architecture created especially for the purpose of detecting anomalies in network traffic is the adaptive neural network.

$$\varepsilon(h) = \dot{P}_{et}(h) - \dot{P}_{dp}(h) \quad (11)$$

Where, $\varepsilon(h)$ is utilized to quantify the mistake and the value of $\varepsilon(h)$ is determined by the difference between $\dot{P}_{et}(h)$ and $\dot{P}_{dp}(h)$. These networks dynamically adjust their structure and parameters based on predefined time intervals, allowing them to adapt to changing network conditions efficiently. EPTANN models can effectively identify anomalies in network traffic patterns, such as sudden spikes or unusual behaviors, which may indicate security threats or system malfunctions.

$$\dot{r}_{et}(h) = \dot{r}_{dp}(y_{f+1}) \quad (12)$$

Where, $\dot{r}_{et}$ define the convergence rate, (h) is define the variable at time h, $\dot{r}_{dp}$ define the stochastic time intervals and y_f define the transmission range. By leveraging their adaptive nature EPTANN can continuously learn and refine their anomaly detection capabilities, improving accuracy and reducing false positives over time.

$$\gamma = \sigma\left|\dot{r}_{dp}(y_{f+1})\right| + \eta_2 \quad (13)$$

Where, η_2 is used to provide a lower limit for γ, $\dot{r}_{dp}$ define the rate of change in another quantity and σ define inversely proportional to time.This approach is crucial for maintaining network security and reliability in today's dynamic and complex environments.

$$\dot{r}_{et}(h)\dot{r}_{dp}(h) \geq 0 \quad (14)$$

Where, $\dot{r}_{et}$ define the convergence rate, (h) is define the variable at time h and $\dot{r}_{dp}$ define the rate of change in another quantity. Useful Predefined Duration The adaptive neural network is one type of neural network architecture designed specifically to identify abnormalities in network traffic.

E. Optimization Using Multi-Agent Cubature Kalman Optimizer

The optimization depends on MACKO[31]. The MACKO utilizes the accuracy of the CKF and the capability of multiple agents to propose a novel method of optimization in complicated systems. In contrast to conventional single-agent techniques, MACKO leverages the combined intelligence of multiple agents, facilitating distributed decision-making and parallel processing. This improves computing efficiency while guaranteeing resilience and flexibility in ever-changing settings. It maintains high accuracy in estimating system states even in the face of non-linearities and uncertainties by integrating Cubature Kalman principles, which makes it a highly useful tool in a variety of fields, such as robotics, banking, and aerospace.The Multi-Agent Cubature Kalman Optimizer (MACKO) was selected due to its proficiency in handling intricate optimization issues in dispersed systems. In contrast to conventional single-agent methods, MACKO is intrinsically robust and scalable since it makes use of numerous agents' abilities to jointly seek for the best solutions. It can manage nonlinearities and uncertainties present in real-world optimization tasks more accurately and effectively by utilizing the Cubature Kalman Filter approach. Due to its distributed nature, this method can also be used in parallel processing, which speeds up convergence and improves overall performance. For this reason, it is a strong option for handling large-scale optimization problems in a variety of industries, including finance,

telecommunications, robotics, and autonomous systems. Figure 2 shows the flow chart of Multi-Agent Cubature Kalman Optimizer.

Step 1: *Initialization*

Early population of MACKO is, initially generated by randomness. Then the initialization is derived in equation

$$x_i^d(0) = randn_i^d + \left[\cup\left(x_d, \bar{x}_d\right)\right] \tag{15}$$

In the d^{th} dimension, where $\bar{x}_d$ is the upper bound of the search space and x_d is the bottom limit, and The agent's number is i^{th}. Furthermore, the solution mistake $p_i^d(0) \in [0,1]$ starting value is produced.

Step 2: *Random Generation*

Parameters for the input are created at random after startup. Their explicit hyperparameter condition determines the best fitness value to choose.

Step 3: *Fitness Function*

The outcome is determined by initialized judgments and random responses. The fitness is then computed using the equation

$$Fitness\ Function = optimizing\ [\lambda_{et} and\ \dot{r}_{dp}] \tag{16}$$

Here λ_{et} represents the increasing accuracy and $\dot{r}_{dp}$ represents the lowering computational time.

Step 4: *Simulate Measurement by Optimizing* λ_{et}

During the investigation phase for the MACKO, researchers go on a journey to learn about its capabilities, limits, and potential applications. This phase begins with a thorough examination of current literature, including research papers, conference proceedings, and patents, to delve into the theoretical foundations and practical applications of MCKO. Researchers undertake simulation studies after thoroughly understanding the algorithm's mathematical formulation and optimization objectives. These simulations allow for the evaluation of MCKO performance across multiple optimization scenarios, including nonlinearities, uncertainties, and limitations, using synthetic data created by mathematical models are given equation

$$z_i^d(t) = \lambda_{et}\left(xp_i^d(t) + \sin\left(randn_i^d \times 2\pi\right) \times \left| xp_i^d(t) - X\_best\_so\_far^d(t)\right|\right) \tag{17}$$

Here the replicated value for agent $z_i^d(t)$ may take is arbitrary location in a locus $\left| xp_i^d(t) - X\_best\_so\_far^d(t)\right|$. A parameter that regulates the pace of change is defined by λ. The stochastic component of the MACKO algorithm is caused by a random element, $randn_i^d \in [0,1]$ in $\sin\left(rand_i^d \times 2\pi\right)$.

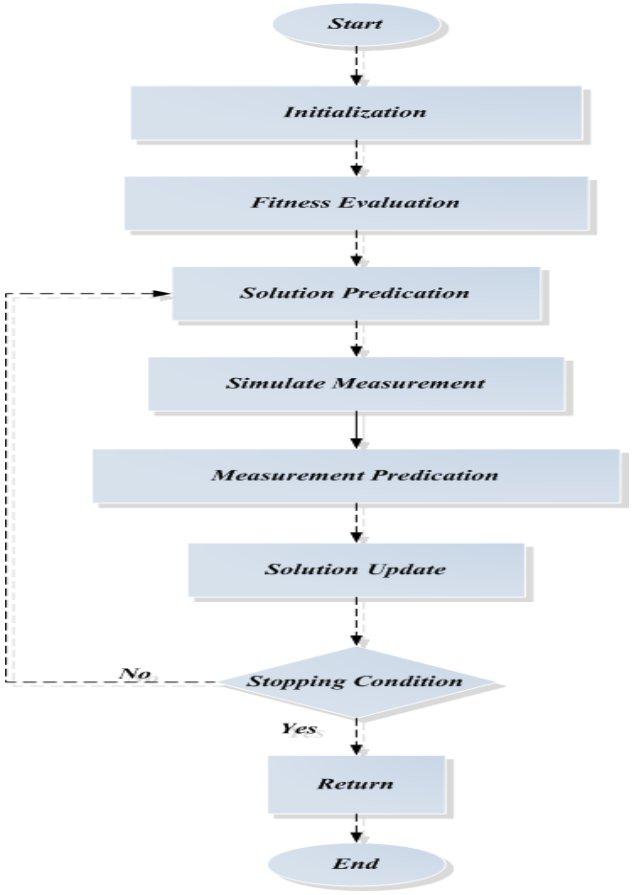

Figure 2: flow chart Multi-Agent Cubature Kalman Optimizer

Step 5: Measurement Prediction by Optimizing $\dot{r}_{dp}$

In the exploitation phase of a MACKO, agents use the collective information gained during the exploration stage to focus their search efforts on attractive regions of the optimization landscape. Using information shared among agents, often in the form of estimated mean and covariance matrices obtained via the Cubature Kalman Filter, each agent narrows its emphasis on areas with potential for optimization. The MCKO controls the range of candidate solution is given in equation

$$T2_{i,j}^{d}(t) = xp_i^d(t) + \dot{r}_{dp}\left[\sqrt{Pp_i^d(t)} - \sqrt{Pp_i^d(t)}\right] \quad (18)$$

Where $T2_{i,j}^{d}(t)$ is the generated cubature point, $\dot{r}_{dp}$ define the stochastic time intervals, $\sqrt{Pp_i^d(t)}$ is the parameter itself, obtained by taking the square root of the average , $xp_i^d(t)$ is determine the predicted solution candidate.

Step 6: Termination

The weight parameter $\lambda_{et} and\ \dot{r}_{dp}$ from EPTANN optimized enhanced by support MACKO, reiteration functions until location information $x = x + 1$ is met. The flow chart for MACKO is given in figure 2.EPTANN is optimized with MACKO for Anomaly Detection, in order to improve the environment by classifying which type of Anomaly Detectionwith greater accuracy.

IV. RESULT AND DISCUSSION

The outcome of a proposed study on the application of a deep learning-based network traffic anomaly detection algorithm. Python is used to implement the suggested approach. Ubuntu 16.04, one Intel Core i7-6900 k processor, four DDR4 total 32 GB RAM, and one CUDA-enabled NVIDIA TITAN XP graphics card were used in the research. Several performance measurements, including Accuracy, Computational Time, F1-Score, Precision, Recall, and ROC, were employed with the Kares Tensor Flow 1.8.0 library and Python 3.6.3-64 bit.

The results of the proposed RA-NTADA-EPTANNmethodology are compared to those of current techniques such as AD-IOTNTT-DNN, NTAD-CNN and NAD-ICE-DCNN techniques respectively.

A. Performances Measures

This is a crucial step for determining the optimal prediction. Performance measures evaluated to assess performance like accuracy, computational time, F1-Score, Precision, Recall, and ROC.

1) Accuracy

Accuracy reflects how closely a measurement aligns with the true value, indicating minimal error or deviation. It is represented in equation,

$$accuracy = \frac{TP + TN}{TP + TN + FN + FP} \quad (19)$$

In this case, TP stands for True Positive rate, TN for True Negative rate, FN for False Negative rate, and FP for False Positive rate.

2) Computation Time

Computation time is the amount of time required by a computer algorithm to do a task .Natural time employs hours and occasionally minutes as units of measurement instead of days.

3) F1- Score

In order to get the F1-Score, which is the mean of recall and accuracy.One needs to first determine both of this metrics. Thus it's give this equation.

$$\text{F1-Score} = \frac{\text{Pr}ecision * \text{Re}call * 2}{\left(\text{Pr}ecision + \text{Re}call\right)} \quad (20)$$

4) Precision

Precision calculates the number of true positives divided by the sum of the true positives and false positives, and the result is by the equation (21),

$$precision = \frac{TN}{FP + TN} \quad (21)$$

5) Recall

The recall of a machine learning model measures how well it can recognize positive examples. Put another way, it measures the likelihood of getting a favorable result. That's provided in equation (22)

$$\text{Re}call = \frac{TP}{\left(TP + FN\right)} \quad (22)$$

6) ROC

It is expressed as the false negative to true positive area ratio by the equation (23)

$$ROC = 0.5 \times \left(\frac{TP}{TP + FP} + \frac{TN}{TN + FP} \right) \quad (23)$$

B. Performance Analysis

Figure 3 to 8 shows the imitation outcomes of suggested RA-NTADA-EPTANNmethod. Proposed RA-NTADA-EPTANNmethod is compared with existing AD-IOTNTT-DNN, NTAD-CNN and NAD-ICE-DCNN method.

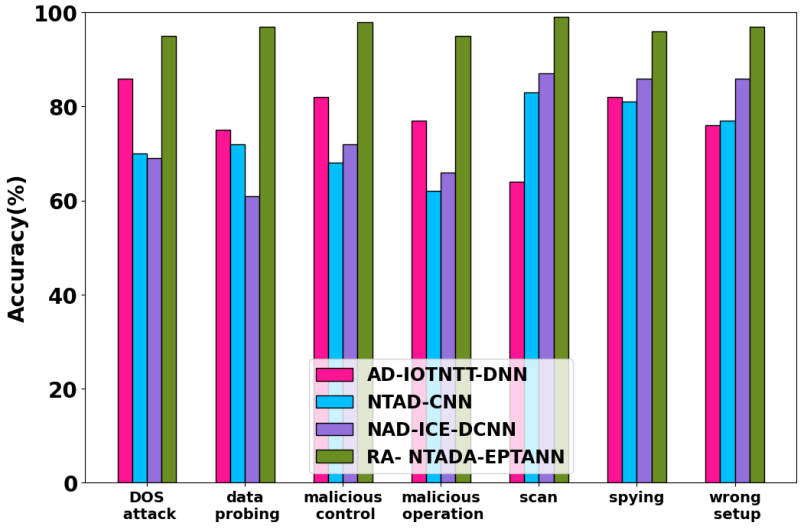

Figure 3: Analysis of Accuracy

Figure 3 shows Analysis of accuracy. The proposed RA-NTADA-EPTANN technique reaches in the range of 16.42%, 23.36%and 19.27% higher accuracy for DOS attack, 33.26%, 17.26% and 20.41% higher accuracy for data probing, 22.36%, 26.42% and 23.27% higher accuracy for malicious control,15.42% and 18.27% higher accuracy for malicious operation, 22.36%, 35.42% and 28.27% higher accuracy for scan, 16.26%, 34.41% and 23.26% higher accuracy for spying, 17.45%, 25.39%and 20.29% higher accuracy for wrong setup Analysis Compared with existing techniques such as AD-IOTNTT-DNN, NTAD-CNN and NAD-ICE-DCNN respectively.

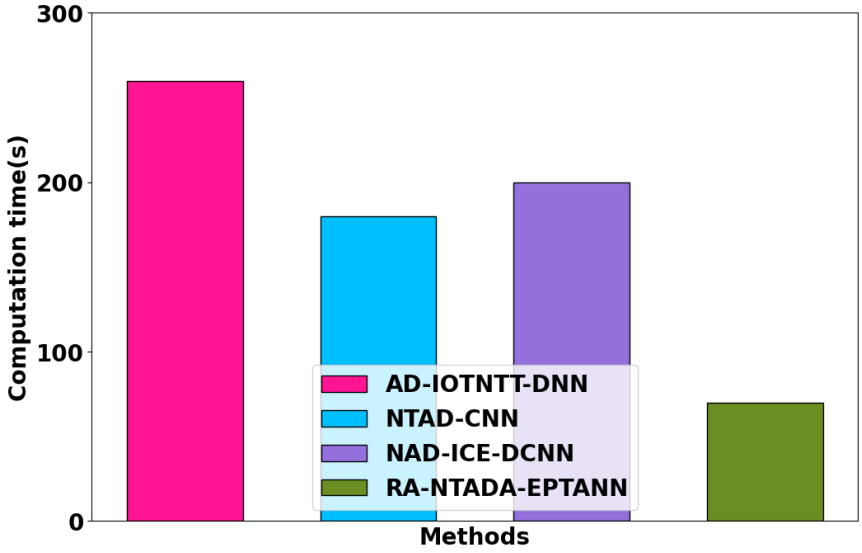

Figure 4: Analysis of Computation Time

Figure 4 depicts the Analysis of Computation Time. Here, proposed RA-NTADA-EPTANNtechnique attains lower computational time of 15.12%, 22.23% and 35.32% analysed with existing AD-IOTNTT-DNN, NTAD-CNN and NAD-ICE-DCNN method respectively. This improvement highlights the efficiency and effectiveness of the new approach in optimizing computational resources for the task at hand.

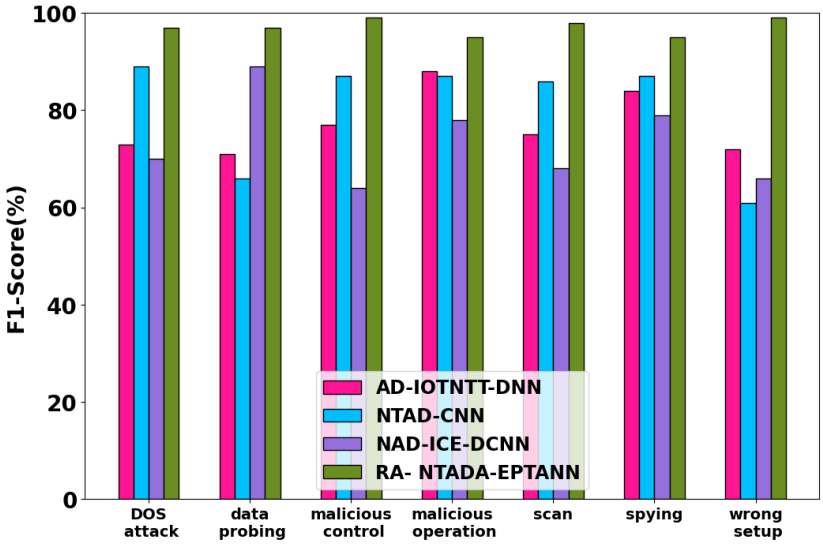

Figure 5: Examining the F1-score

Figure 5 showsExamining the F1-score. The proposed RA-NTADA-EPTANN technique reaches in the range of 18.11%, 22.29%and 28.17% higher F1-Score for DOS attack, 33.26%, 17.26% and 20.41% higher F1-Score for data probing, 22.36%, 19.36%and 23.27% higher F1-Score for malicious control,15.42% and 18.27% higher F1-Score for Normal, 21.29%, 20.41%, and 27.28% higher F1-Score for malicious operation, 19.30%, 21.32% and 24.36% higher F1-Score for scan,16.56%, 27.31%and 24.23% higher F1-Score for spying, 32.22%, 20.24% and 26.12% higher F1-Score for wrong setup Analysis Compared with existing techniques such as AD-IOTNTT-DNN, NTAD-CNN and NAD-ICE-DCNN respectively.

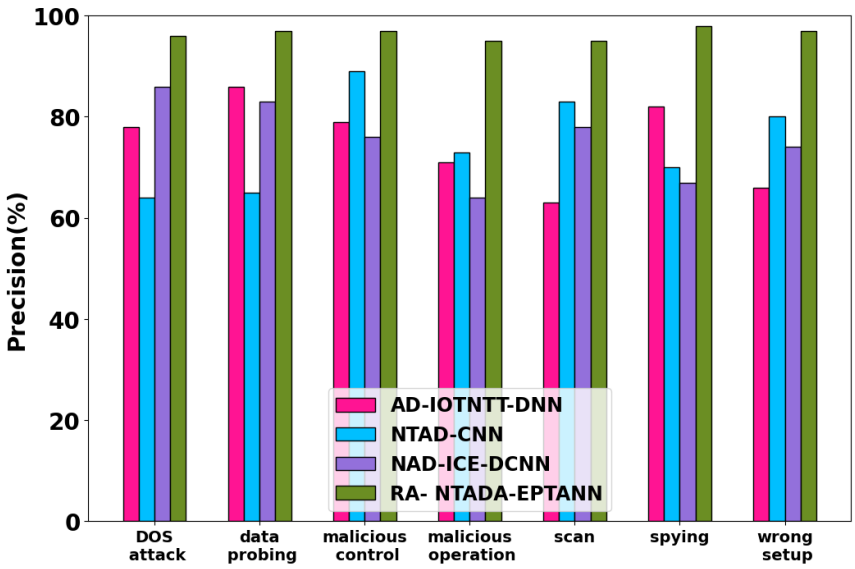

Figure 6: Analysis of Precision

Figure 6 shows Analysis of precision. The proposed RA-NTADA-EPTANN technique reaches in the range of 17.45%, 25.39%and 20.29% higher precision for DOS attack, 35.29%, 19.28% and 24.42% higher precision for data probing, 24.39%, 18.48% and 20.28% higher precision for malicious control,19.40% and 20.27% higher precision for malicious operation, 26.36%, 39.44% and 24.28% higher precision for scan, 18.26%, 38.42% and 25.30% higher precision for spying and 17.26% and 20.41% higher precision for wrong setup Associated with current techniques such as AD-IOTNTT-DNN, NTAD-CNN and NAD-ICE-DCNN respectively.

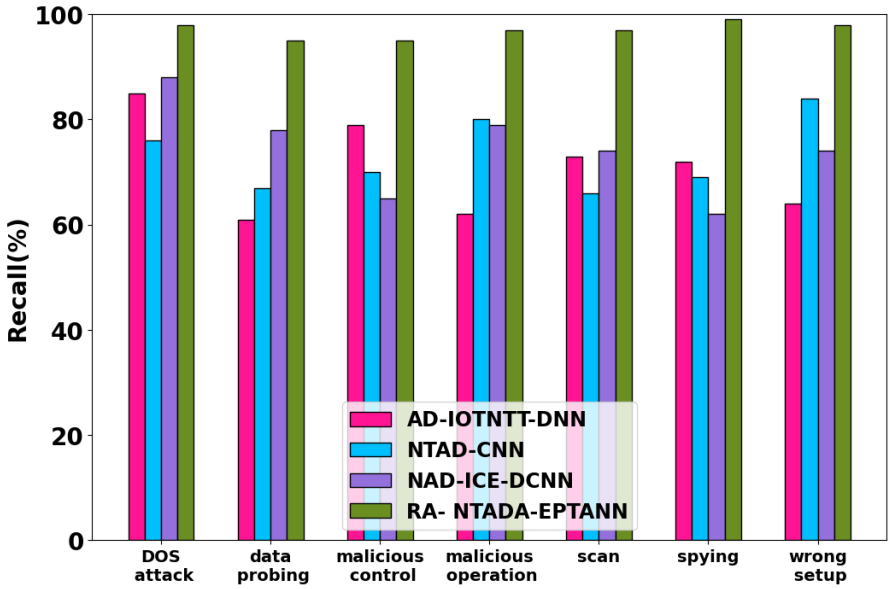

Figure 7: Analysis of Recall

Figure 7 shows Analysis of Recall. The proposed RA-NTADA-EPTANN technique reaches in the range of 18.42%, 27.36%and 18.27% higher Recall for DOS attack, 34.26%, 18.26% and 21.41% higher Recall for data probing, 23.36%, 16.42% and 19.27% higher Recall for malicious control,17.42% and 19.27% higher Recall for malicious operation, 23.36%, 37.42% and 29.27% higher Recall for scan, 17.26%, 38.41% and 29.26% higher Recall for spying and 26.36%, 39.44% and 24.28% higher Recall for wrong setup Compared with existing techniques such as AD-IOTNTT-DNN, NTAD-CNN and NAD-ICE-DCNN respectively.

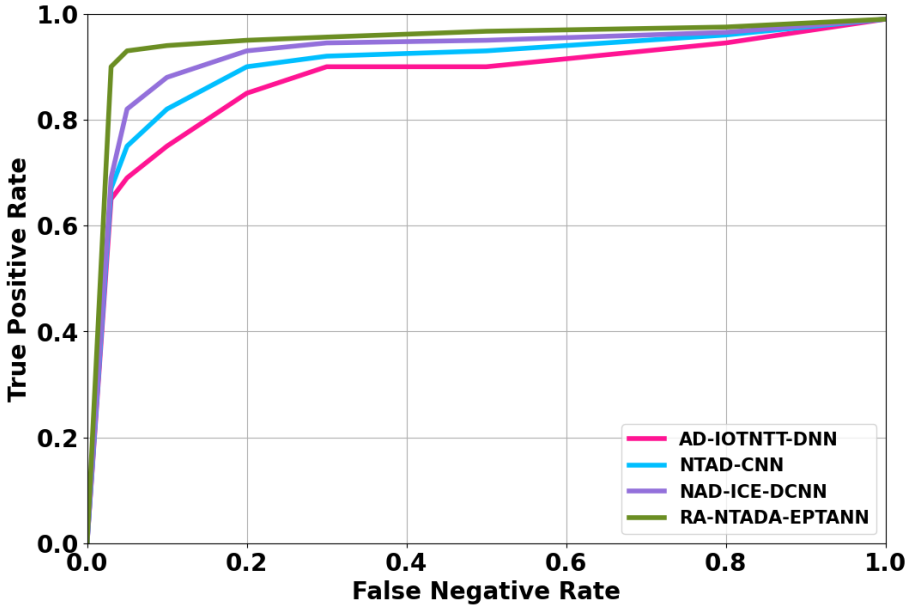

Figure 8: Analysis of ROC

Figure 8 depicts the Analysis of ROC curve. This curve plots 4 parameters. The proposed RA-NTADA-EPTANNmethod attains 0.35%, 0.42%, and 0.41% higher ROC estimated to the existing method such anAD-IOTNTT-DNN, NTAD-CNN and NAD-ICE-DCNN method respectively.

C. Discussion

This study develops the RA-NTADA-EPTANNmodels initial step toward Network Traffic Anomaly Detection. After that, data cleaning is used to feed the gathered data into the pre-processing CPSF, addressing noisy and missing values in the dataset. The DOS attack, data probing, malicious control, malicious operation, scan, surveillance, and incorrect setup are all classified by the EPTANN anomaly detection. The optimized using MACKO. The performance of the developed as AD-IOTNTT-DNN, NTAD-CNN and NAD-ICE-DCNN was assessed using the following metrics: Accuracy, Computational Time, F1-Score, Precision, Recall and ROC.

The performance of the suggested work compared to current techniques such as AD-IOTNTT-DNN, NTAD-CNN and NAD-ICE-DCNN, the Proposed RA-NTADA-EPTANNmethod attain 15.12%, 22.23% and 35.32% higher computational time analysed method has higher evaluation metrics for recall and accuracy. Consequently, the suggested method more successfully and efficiently classifies DOS attacks, data probing, malicious control, malicious operation, scan, surveillance, and incorrect setup.

V. CONCLUSION

In present study, RA-NTADA-EPTANN was successfully implemented. The proposed RA-NTADA-EPTANNexecuted in python working platform. Performance of RA-NTADA-EPTANNapproach 15.12%, 22.23% and 35.32% higher computational time is compared with existing methods such as AD-IOTNTT-DNN, NTAD-CNN and NAD-ICE-DCNN method. They require to perform traffic statistics, follow moving items, and investigate the suggested model's capacity for object identification and classification on different types of objects in the future.

REFERENCE

[1] Fosić, I., Žagar, D., Grgić, K. and Križanović, V., 2023. Anomaly detection in NetFlow network traffic using supervised machine learning algorithms. *Journal of industrial information integration*, p.100466.

[2] Xu, H., Sun, Z., Cao, Y. and Bilal, H., 2023. A data-driven approach for intrusion and anomaly detection using automated machine learning for the Internet of Things. *Soft Computing*, *27*(19), pp.14469-14481.

[3] Zuhairi, M.F., Ali, S.M., Shahid, Z., Alam, M.M. and Su'ud, M.M., 2024. Realtime Feature Engineering for Anomaly Detection in IoT based MQTT Networks. *IEEE Access*.

[4] Arul, U., Arun, V., Rao, T.P., Baskaran, R., Kirubakaran, S. and Hussan, M.T., 2024. Effective Anomaly Identification in Surveillance Videos Based on Adaptive Recurrent Neural Network. *Journal of Electrical Engineering & Technology*, pp.1-13.

[5] Ullah, F., Ullah, S., Srivastava, G. and Lin, J.C.W., 2024. IDS-INT: Intrusion detection system using transformer-based transfer learning for imbalanced network traffic. *Digital Communications and Networks*, *10*(1), pp.190-204.

[6] Chen, Z., Jia, D., Sun, Y., Yang, L., Jin, W. and Liu, R., 2024. Univariate Time Series Anomaly Detection Based on Hierarchical Attention Network. *Tsinghua Science and Technology*, *29*(4), pp.1181-1193.

[7] Tarazi, H., Sutton, S., Olinjyk, J., Bond, B. and Rrushi, J., 2024. A watchdog model for physics-based anomaly detection in digital substations. *International Journal of Critical Infrastructure Protection*, *44*, p.100660.

[8] Sharma, P., Sharma, S.K. and Dani, D., 2024. Edge-assisted federated learning for anomaly detection in diverse IoT network. *International Journal of Information Technology*, pp.1-11.

[9] Nallappan, M. and Velswamy, R., 2024. Exploring deep learning-based content-based video retrieval with Hierarchical Navigable Small World index and ResNet-50 features for anomaly detection. *Expert Systems with Applications*, *247*, p.123197.

[10] Akagic, A. and Džafić, I., 2024. Enhancing smart grid resilience with deep learning anomaly detection prior to state estimation. *Engineering Applications of Artificial Intelligence*, *127*, p.107368.

[11] Mishra, S. and Jabin, S., 2024. Anomaly detection in surveillance videos using deep autoencoder. *International Journal of Information Technology*, *16*(2), pp.1111-1122.

[12] Nixon, C., Sedky, M., Champion, J. and Hassan, M., 2024. SALAD: A split active learning based unsupervised network data stream anomaly detection method using autoencoders. *Expert Systems with Applications*, p.123439.

[13] Cheng, L. and Sun, K., 2024. Research on intelligent vehicle Traffic Flow control algorithm based on data mining. *International Journal of Intelligent Networks*.

[14] Salam, A., Abrar, M., Amin, F., Ullah, F., Khan, I.A., Alkhamees, B.F. and Alsalman, H., 2024. Securing smart manufacturing by integrating anomaly detection with zero-knowledge proofs. *IEEE Access*.

[15] Saheel, S., Alvi, A., Ani, A.R., Ahmed, T. and Uddin, M.F., 2024. Semi-supervised, Neural Network based approaches to face mask and anomaly detection in surveillance networks. *Journal of Network and Computer Applications*, *222*, p.103786.

[16] Almotairi, A., Atawneh, S., Khashan, O.A. and Khafajah, N.M., 2024. Enhancing intrusion detection in IoT networks using machine learning-based feature selection and ensemble models. *Systems Science & Control Engineering*, *12*(1), p.2321381.

[17] Maseno, E.M. and Wang, Z., 2024. Hybrid wrapper feature selection method based on genetic algorithm and extreme learning machine for intrusion detection. *Journal of Big Data*, *11*(1), p.24.

[18] Chen, X., Wang, P., Yang, Y. and Liu, M., 2024. Resource-Constraint Deep Forest Based Intrusion Detection Method in Internet of Things for Consumer Electronic. *IEEE Transactions on Consumer Electronics*.

[19] Golling, T., Kasieczka, G., Krause, C., Mastandrea, R., Nachman, B., Raine, J.A., Sengupta, D., Shih, D. and Sommerhalder, M., 2024. The interplay of machine learning-based resonant anomaly detection methods. *The European Physical Journal C*, *84*(3), p.241.

[20] Jiao, Q. and Mhamdi, L., 2024, February. Deep Learning based Intrusion Detection for IoT Networks. In *2024 Global Information Infrastructure and Networking Symposium (GIIS)* (pp. 1-6). IEEE.

[21] Reddy, D.K., Behera, H.S., Nayak, J., Vijayakumar, P., Naik, B. and Singh, P.K., 2021. Deep neural network based anomaly detection in Internet of Things network traffic tracking for the applications of future smart cities. *Transactions on Emerging Telecommunications Technologies*, *32*(7), p.e4121.

[22] Sheng, S. and Wang, X., 2023. Network traffic anomaly detection method based on chaotic neural network. *Alexandria Engineering Journal*, *77*, pp.567-579.

[23] Vibhute, A.D. and Nakum, V., 2024. Deep learning-based network anomaly detection and classification in an imbalanced cloud environment. *Procedia Computer Science*, *232*, pp.1636-1645.

[24] Ajila, S.A., Lung, C.H. and Das, A., 2022. Analysis of error-based machine learning algorithms in network anomaly detection and categorization. *Annals of Telecommunications*, *77*(5), pp.359-370.

[25] Hooshmand, M.K. and Hosahalli, D., 2022. Network anomaly detection using deep learning techniques. *CAAI Transactions on Intelligence Technology*, *7*(2), pp.228-243.

[26] Patil, R., Biradar, R., Ravi, V., Biradar, P. and Ghosh, U., 2022. Network traffic anomaly detection using PCA and BiGAN. *Internet Technology Letters*, *5*(1), p.e235.

[27] Yao, C., Yang, Y., Yin, K. and Yang, J., 2022. Traffic anomaly detection in wireless sensor networks based on principal component analysis and deep convolution neural network. IEEE Access, 10, pp.103136-103149.

[28] Qiang, X., Xue, R. and Zhu, Y., 2024. Confidence partitioning sampling filtering. *EURASIP Journal on Advances in Signal Processing*, *2024*(1), p.24.

[29] Anaraki, M.V. and Farzin, S., 2023. Humboldt Squid Optimization Algorithm (HSOA): A Novel Nature-Inspired Technique for Solving Optimization Problems. *IEEE Access*, *11*, pp.122069-122115.

[30] Qi, Z., Ning, Y., Xiao, L., Wang, Z. and He, Y., 2024. Efficient Predefined-Time Adaptive Neural Networks for Computing Time-Varying Tensor Moore–Penrose Inverse. *IEEE Transactions on Neural Networks and Learning Systems*.

[31] Musa, Z., Ibrahim, Z. and Shapiai, M.I., 2024. Multi-Agent cubature Kalman optimizer: A novel metaheuristic algorithm for solving numerical optimization problems. *International Journal of Cognitive Computing in Engineering*, *5*, pp.140-152.